Лабораторная работа № 1.1

Тема: Разработка текстовой RPG-игры. Инкапсуляция

#include <iostream>

#include <string>

class Character {

private:

std::string name; // Приватное поле: имя персонажа

int health; // Приватное поле: уровень здоровья

int attack; // Приватное поле: уровень атаки

int defense; // Приватное поле: уровень защиты

public:

// Конструктор для инициализации данных

Character(const std::string& n, int h, int a, int d)

: name(n), health(h), attack(a), defense(d) {}

// Метод для получения уровня здоровья

int getHealth() const {

return health;

}

// Метод для вывода информации о персонаже

void displayInfo() const {

std::cout << "Name: " << name << ", HP: " << health

<< ", Attack: " << attack << ", Defense: " << defense << std::endl;

}

// Метод для атаки другого персонажа

void attackEnemy(Character& enemy) {

int damage = attack - enemy.defense;

if (damage > 0) {

enemy.health -= damage;

std::cout << name << " attacks " << enemy.name << " for " << damage << " damage!" << std::endl;

} else {

std::cout << name << " attacks " << enemy.name << ", but it has no effect!" << std::endl;

}

}

// Метод лечения персонажа

void heal(int amount) {

health += amount;

if (health > 100) {

health = 100;

}

std::cout << name << " heals for " << amount << " HP. Current HP: " << health << std::endl;

}

// Метод получения урона

void takeDamage(int amount) {

health -= amount;

if (health < 0) {

health = 0;

}

std::cout << name << " takes " << amount << " damage. Current HP: " << health << std::endl;

}

};

int main() {

// Создаем объекты персонажей

Character hero("Hero", 100, 20, 10);

Character monster("Goblin", 50, 15, 5);

// Выводим информацию о персонажах

hero.displayInfo();

monster.displayInfo();

// Герой атакует монстра

hero.attackEnemy(monster);

monster.displayInfo();

// Проверка лечения

monster.heal(30);

monster.displayInfo();

// Проверка получения урона

monster.takeDamage(60);

monster.displayInfo();

// Проверка значений, превышающих норму

Character test("Test", 30, 0, 0);

test.displayInfo();

test.takeDamage(50);

test.displayInfo();

test.heal(30);

test.displayInfo();

test.heal(150);

test.displayInfo();

return 0;

}